Energy-Aware Blockchain Resource Allocation Algorithm with Deep Reinforcement Learning for Trusted Authentication

Lifang Gao[1], Xiaotao Zhang[2], Tingfeng Liu[2], Huifeng Yang[1], Boxian Liao[3(✉)], and Jing Guo[2]

[1] State Grid Hebei Information & Telecommunication Branch, Shijiazhuang, China
[2] Aostar Information Technologies Co., Ltd. 2688 Xiyuan Dadu, South of Modern Industrial Port, Chengdu, China
[3] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
boxian_liao@bupt.edu.cn

Abstract. Internet of things (IoT) technology is in continuous development, and the access of the IoT power terminal is facing various security threats such as data tampering and malicious attacks. Thus, we propose a blockchain-based edge-terminal collaborative resource allocation architecture to solve these security problems, which places the terminal trusted authentication data on the blockchain to realize the security of the terminal authentication information. Since the mining process of the blockchain system will generate a large number of computing intensive tasks, this paper establishes an energy-oriented blockchain mining task offloading model, and proposes the energy-aware blockchain resource allocation (EABRA) algorithm with deep reinforcement learning (DRL) to jointly optimize the offloading decision and transmission power allocation decision. Finally, the simulation results show that the EABRA algorithm can save 68.87% energy consumption than the Random algorithm, which verifies the correctness and feasibility of the scheme.

Keywords: Internet of things · Edge-terminal collaborative · Trusted authentication · Task offloading

1 Introduction

Recently, the power terminals of the Internet of things (IoT) present a variety of characteristics, and most power terminals are more vulnerable to be attacked due to their low security. The traditional terminal authentication mechanism faces many security problems such as data leakage and data tampering. And the centralized authority solution is not suitable for the authentication and access of highly distributed IoT devices [1]. The distributed characteristics of power terminals need a self-protection mechanism that does not depend on the central authority [2]. In order to realize the trusted authentication

M. Cheng et al. (Eds.): SmartGIFT 2020, LNICST 373, pp. 93–103, 2021.
https://doi.org/10.1007/978-3-030-73562-3_8

of power terminal and reduce the security risk of terminal access, we use the security and tamper-resistant of blockchain technology, and use mobile intelligent terminal as the key carrier of identity authentication and electronic signature. The core data is placed on the blockchain, the blockchain is used to realize the safe storage of authentication information, and the smart contract is used to realize the safe and automated execution of the authentication process [3].

Based on the analysis of the research status at home and abroad, it can be seen that many scholars have made a lot of research achievements on blockchain-based trusted authentication methods. Neisse et al. [4] proposed a blockchain-based platform to enhance the transparency and traceability of network security authentication information and realize the trusted exchange of IoT security authentication information. Cui et al. [5] proposed a blockchain-based multi-sensor network authentication scheme for the IoT, which builds a blockchain network between different types of nodes to achieve mutual authentication of node identities in different communication scenarios. Thakker et al. [6] proposed a data management system based on blockchain technology to ensure the security of access terminals, and the authenticated terminals can ensure data integrity and provide trusted data storage. The non-tamper ability of blockchain can well guarantee the trust of the system. However, the application of blockchain technology is limited by the mining process, that is, miners (mobile terminals) are required to complete computing intensive tasks, which puts forward extremely high requirements for the computing capacity of the terminals [7].

We introduce mobile edge computing (MEC) technology to solve the problem of the limited computing capacity of the blockchain system. In order to achieve the integrity and validity of authentication information, we then propose a blockchain-based edge-terminal collaborative resource allocation architecture to ensure the trust of terminal authentication.

2 System Model

2.1 Application Model

As shown in Fig. 1, the blockchain system is located at the end layer, and we store the trusted authentication information of the terminals in the block. The MEC server is located in the edge layer, and the edge node can be connected with the base station. Let $\mathbf{N} = \{1, 2, \ldots, \mathbf{N}\}$ represent the set of base stations and $\mathbf{M} = \{1, 2, \ldots, \mathbf{M}\}$ represent the set of mobile terminals. Suppose that each mobile terminal can perform the mining task K_n of the blockchain system as a miner x_n, and the mobile terminal can access its nearest edge node for collaborative task offloading. We use f_n and f_m (CPU cycles/s) to represent the computing capability of the edge node and the mobile terminal respectively. For miner x_n, we use $<D_n, \tau_n, X_n>$ to denote the data size D_n (bit), the completion time τ_n (second), and the computing intensity X_n (CPU cycles/bit) of the mining task K_n. The symbols used in this paper are summarized in Table 1.

Table 1. Notation definitions.

Symbol	Definition
N	The set of base stations
M	The set of mobile terminals
D_n	The data size of the task
τ_n	The completion time of the task
X_n	The computing intensity of the task
f_m	The computing capability of mobile terminal
f_n	The computing capability of edge node
ζ_m	The computing energy efficiency coefficient of mobile terminal
ζ_n	The computing energy efficiency coefficient of edge node
P_s	The static circuit power
$P_n(t)$	The transmission power between mobile terminal and edge node
$g_n(t)$	The channel gain between mobile terminal and edge node
$\Gamma_{\min}$	The minimum computing power required by blockchain system

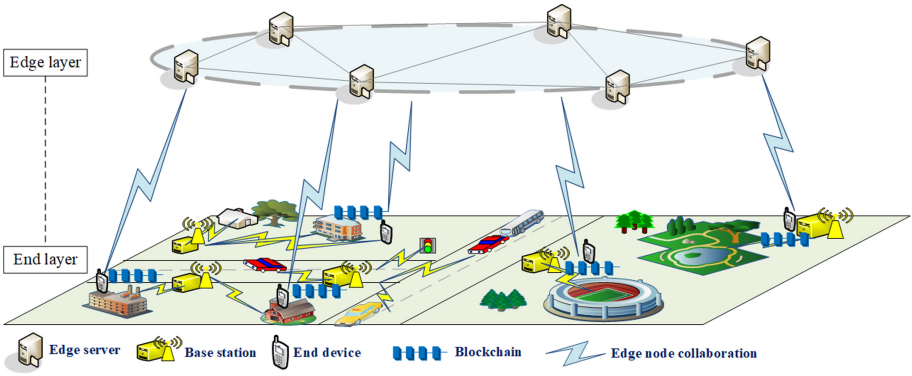

Fig. 1. The blockchain-based edge-terminal collaborative resource allocation architecture.

2.2 MEC Model

In the MEC system, we assume that all mobile terminals and edge nodes have computing capability to perform the mining task of blockchain. Due to the limited computing capability of mobile terminal, it may not be able to handle a large number of computing intensive tasks, so we use MEC server to solve this problem. The MEC server has powerful computing resources. We can offload the computing tasks of the terminal to the edge node for collaborative computing, so as to improve the computing speed of the system. Therefore, we consider two different calculation modes, and let $d_n(t) \in \{0, 1\}$ represent computing offloading decision of miner x_n. $d_n(t) = 0$ indicates that the miner

x_n selects mode 0, that is, the mobile terminal performs computing task. $d_n(t) = 1$ indicates that the miner x_n chooses mode 1, that is, the mobile terminal offloads the computing task to the edge node for computing.

Mobile Terminal Computing
We analyze the performance of miner x_n when performing computing task on the mobile terminal in this mode. The delay of local computing includes the time for miner x_n to complete the task. We use f_m to represent the computing capability of the mobile terminal, that is, the time for miner x_n to perform the mining task K_n can be expressed as:

$$T_m(t) = \frac{D_n X_n}{f_m} \tag{1}$$

Let χ represent the number of CPU cycles required by the mobile terminal to process 1-bit computing task. The computing rate of the mobile terminal in this mode can be denoted as:

$$r_m(t) = \frac{f_m}{\chi} \tag{2}$$

We use ζ_m to represent the computing energy efficiency coefficient of mobile terminal, so the energy consumption of the system can be expressed as:

$$E_m(t) = \zeta_m (f_m)^3 T_m(t) \tag{3}$$

Edge Node Collaborative Computing
The time taken by miner x_n to upload K_n to the edge node can be denoted as:

$$T_n^u(t) = \frac{D_n}{r_n(t)} \tag{4}$$

where $r_n(t)$ is the transmission rate from the mobile terminal to the edge node, we let B denote the channel bandwidth, $g_n(t)$ and $P_n(t)$ denote the channel gain and transmission power between the mobile terminal and the edge node respectively.

$$r_n(t) = B \cdot \log_2(1 + \frac{P_n(t) g_n(t)}{\sigma_n^2(t)}) \tag{5}$$

where $\sigma_n^2(t)$ is the noise power between the mobile terminal and the edge node. We use f_n to represent the computing capability of the edge node, so the time for the edge node to complete the mining task K_n can be expressed as:

$$T_n^e(t) = \frac{D_n X_n}{f_n} \tag{6}$$

We assuming that the number of CPU cycles in the task buffer of the edge server is Q_n, then the queuing delay of miner x_n can be denoted by:

$$T_n^q(t) = \frac{Q_n}{f_n} \tag{7}$$

We use ζ_n to represent the computing energy efficiency coefficient of edge node, and P_s to represent the static circuit power. In this mode, the energy consumption of the system can be denoted as:

$$E_n(t) = P_n(t)T_n^u(t) + \zeta_n(f_n)^3 T_n^e(t) + P_s T_n^q(t) \tag{8}$$

Additionally, we use $E_{tot,n}(t)$ to represent the total energy consumption of the system, which can be denoted as:

$$E_{tot,n}(t) = (1 - d_n(t))E_m(t) + d_n(t)E_n(t) \tag{9}$$

2.3 Blockchain System

We can put the authentication information of mobile terminals in the blockchain system. And we choose some nodes with high voting rate as consensus nodes of blockchain to participate in block generation and verification, which can improve the system performance [8]. In the consensus process of the blockchain system, we use the Delegated Byzantine Fault Tolerance consensus mechanism. The nodes are divided into agent nodes and ordinary nodes. The agent nodes have the right to keep accounts. The ordinary nodes can see the consensus process and synchronize the ledger information. The number of votes for consensus nodes depends on the number of stakes and available computing resources. The available computing resources refer to the remaining computing resources of the node after the offloading task is processed.

We use $\mathbf{S}(t) = \{S_1(t), S_2(t), ..., S_n(t)\}$ and $\mathbf{\Gamma}(t) = \{\Gamma_1(t), \Gamma_2(t), ..., \Gamma_n(t)\}$ to represent the set of the stake and available computing resources. Assume that there is a data buffer in the edge server to store offloading tasks that have arrived but have not yet been executed. Additionally, we use Ω_{m} and $\Gamma_{\min}$ to represent the total computing capability of the edge server and the minimum computing resources required by the blockchain system, respectively. The computing resources available for the blockchain system in the edge server can be denoted as:

$$\Gamma_n(t) = \max\{\Omega_{\mathrm{m}} - \Omega_n(t), \Gamma_{\min}\} \tag{10}$$

Let ρ_n to represent the processing density, so the dynamics of the system processing queue can be defined as:

$$\Omega_n(t+1) = \max\{\Omega_n(t) - f_m + \rho_n r_{tot}(t), 0\} \tag{11}$$

where $r_{tot}(t)$ is the total computing rate of the system, and it can be denoted as:

$$r_{tot}(t) = (1 - d_n(t))r_m(t) + d_n(t)r_n(t) \tag{12}$$

3 Problem Formulation

We model the optimization problem as a Markov Decision Process (MDP) to obtain the optimal resource allocation strategy. Due to the dynamic characteristics of the system, we propose an algorithm based on deep reinforcement learning (DRL) asynchronous advantage actor-critic (A3C) to solve this problem. We use tuple $<\mathbf{S}, \mathbf{A}, Pr, r>$ to define the Markov decision process, where $\mathbf{S}$ is the state space, $\mathbf{A}$ is the action space, Pr is the state transition probability, and r is the reward function.

3.1 Optimization Objective

In order to realize the reasonable resource allocation of the system, we propose an optimization problem to minimize the total energy consumption of the system, and optimize the offloading decision and transmission power allocation decision. We express the optimization problem as follows:

$$\begin{aligned} \min \ & \sum_{t=0}^{T-1} \sum_{n=1}^{N} E_{tot,n}(\mathrm{t}) \\ \text{s.t.} \ & T_{tot,n} < \tau_n && \text{C1} \\ & 0 \le P_{tot,n}(\mathrm{t}) \le P && \text{C2} \\ & 0 \le f_m < f_n \le f_{\max} && \text{C3} \\ & d_n(\mathrm{t}) \in \{0, 1\} && \text{C4} \end{aligned} \tag{13}$$

In order to meet the requirements of task delay, constraint C1 ensures that the total time for the system to complete the mining task does not exceed the completion time τ_n. Constraint C2 means that the sum of transmission power does not exceed the total power P. Constraint C3 ensures that all CPU frequencies are non-negative and finite. Constraint C4 ensures that the task offloading decision is effective.

3.2 Problem Transformation

State Space

We express the state space as the combination of channel condition $\mathbf{G}(t) = \{g_n(t), g_{n,\mathrm{k}}(t)\}$ and available computing resource $\mathbf{\Gamma}(t) = \{\Gamma_1(t), \Gamma_2(t), ..., \Gamma_n(t)\}$ of the MEC server:

$$\mathbf{S}(t) \triangleq \{\mathbf{G}(t), \mathbf{\Gamma}(t)\} \tag{14}$$

Action Space

We use $\mathrm{A}(t) = [\mathbf{d}(t), \mathbf{P}(t)]$ to define the action space. We define the task offloading decision $\mathbf{d}(t)$ and the transmission power allocation decision $\mathbf{P}(t)$ as:

$$\mathbf{d}(t) \triangleq \{d_1(t), d_2(t), ..., d_N(t)\} \tag{15}$$

$$\mathbf{P}(t) \triangleq \{P_1(t), P_2(t), ..., P_N(t)\} \tag{16}$$

State Transition Probability

After performing an action, the probability of leaving the state $s(t)$ to the next state $s(t+1)$ can be defined as:

$$Pr(s(t+1)|s(t), a(t)) = \int_{s^t}^{s^{t+1}} f(s(t), a(t), s)\, ds \tag{17}$$

where f is the state transition probability density function.

Reward Function
The reward function in this system can be defined as:

$$r_s = \begin{cases} R(t), & \text{if } C1 - C4 \text{ are satisfied} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $R(t) = \frac{1}{\sum_{n=1}^{N} E_{tot,n}(t)}$.

3.3 Problem Solution

A3C is a parallel implementation of deep reinforcement learning asynchronous method [9]. A3C algorithm is to create multiple parallel environments on a machine, put actor and critic in multiple different threads for training and assign tasks, and update the parameters of local network to the global network each single thread completes the learning, and acquire the comprehensive learning of updated parameters from the global network regularly. Each thread will learn from the environment independently and explore different strategies in parallel. Therefore, we use A3C to explore the optimal offloading decision of edge-terminal collaboration.

4 Simulation Results and Analysis

In order to evaluate the performance of the EABRA algorithm under different parameters, we use TensorFlow 2.0 based on Python 3.7 for simulation, and establish a model composed of MEC system and blockchain system, in which the network coverage radius is about 500 m. The main simulation parameters are summarized in Table 2.

Table 2. Simulation parameters.

Parameter	Value
The computing capability of mobile terminal f_m	1 GHz
The computing capability of edge node f_n	2.4 GHz
The processing density χ	737.5 cycles/bit
The noise power density N_0	−174 dBm/Hz
The bandwidth B	180 KHz
The learning rate for actor network η_a	0.001
The learning rate for critic network η_c	0.01
The static circuit power P_s	0.05 W
The data size of the task D_n	0.42 MB

As shown in Fig. 2, we can use TensorBoard, the built-in module of TensorFlow, to see the visualization of A3C algorithm architecture. We can see that the A3C architecture consists of a global network and eight worker agents. The A3C algorithm starts from building a global network.

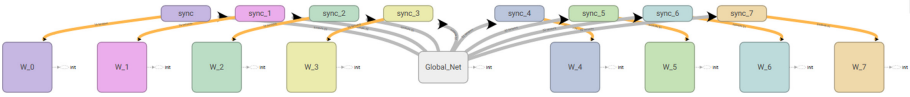

Fig. 2. Visualization of DRL algorithm based on TensorBoard.

Figure 3 shows the internal structure of a worker agent. Each worker agent has actor network and critic network, and can interact with the surrounding environment. After the interaction, worker agents will update the global network parameters according to their own network parameters.

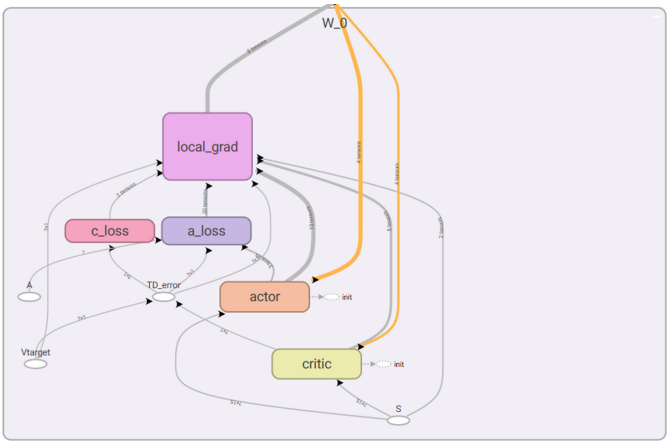

Fig. 3. Visualization of the worker agent based on TensorBoard.

In order to verify the feasibility of the EABRA algorithm, we compare it with the Random (mobile terminals perform actions randomly). In Fig. 4, we compare the loss functions of the two algorithms, and we can see that the EABRA algorithm converges faster, and the performance of the Random is poor, which can hardly converge.

Figure 5 shows the impact of the computing capability of edge node on the total energy consumption. It can be seen that for all schemes, the total energy consumption increases with the increase of f_n. This is because as the CPU frequency of the edge node increases, although the calculation rate will increase, the communication overhead will also increase, which will lead to additional energy consumption, thus increasing the total energy consumption of the system. Figure 6 shows the effect of transmission power P

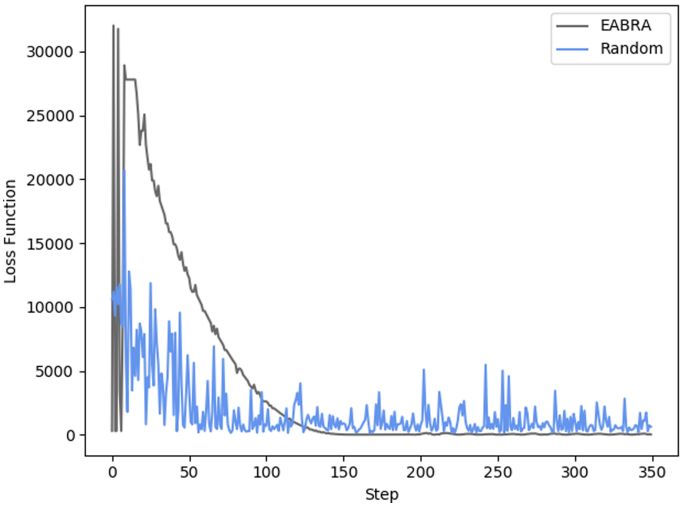

Fig. 4. The change curve of loss function based on A3C algorithm.

on the average reward. It can be seen that the performance of EABRA is always the best, because we jointly optimize the task offloading decision and the transmission power allocation decision, and find the optimal resource allocation scheme base on the DRL algorithm.

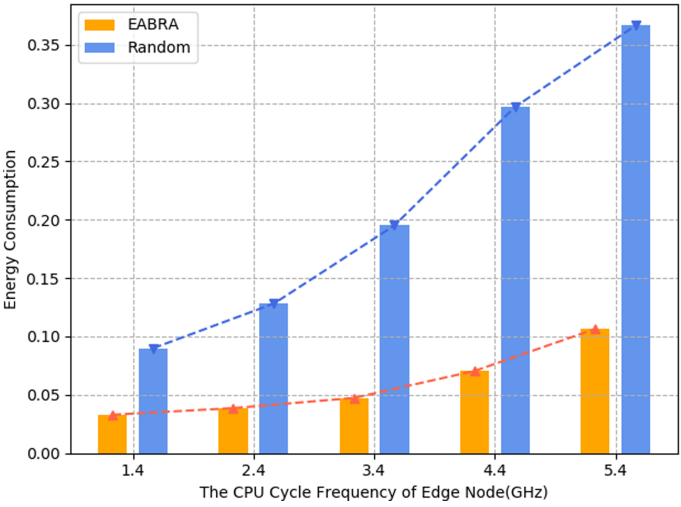

Fig. 5. The impact of f_n on total energy consumption of the system.

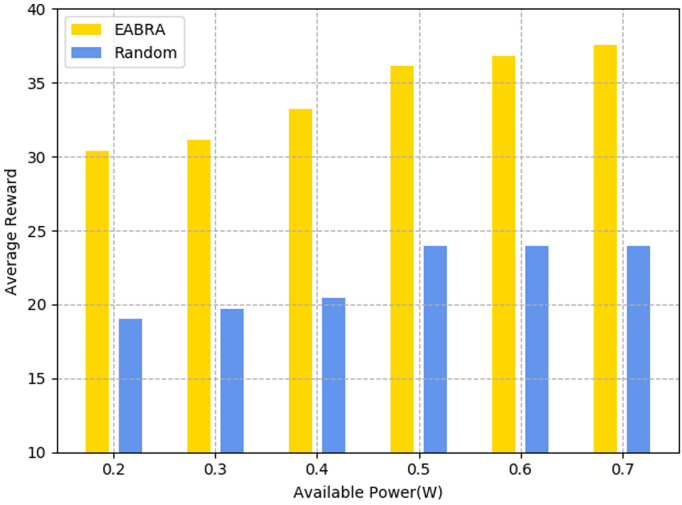

Fig. 6. The impact of P on average reward of the system.

5 Conclusion

This paper proposes a blockchain-based edge-terminal collaborative resource allocation architecture, including MEC system and blockchain system. Local mobile terminal computing mode and edge node collaborative computing mode are used to process the mining task of blockchain system, so as to ensure the trust of terminal authentication information. In order to optimize the total energy consumption of the system, we model the problem as an MDP problem, jointly optimize the offloading decision and power allocation decision, and propose the EABRA algorithm with deep reinforcement learning for solution. The simulation results clearly show the superiority of EABRA algorithm. Under different parameter settings, the EABRA algorithm has faster convergence speed and better performance.

Acknowledgment. This work is supported by the Science and Technology Project of State Grid Corporation of China: Research on Key Technologies of dynamic identity security authentication and risk control in power business (SGHEXT00YJJS1900050).

References

1. Roman, R., Zhou, J., Lopez, J.: On the features and challenges of security and privacy in distributed internet of things. Comput. Netw. **57**(10), 2266–2279 (2013)
2. Rashid, M.A., Pajooh, H.H.: A security framework for iot authentication and authorization based on blockchain technology. In: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, pp. 264–271 (2019)

3. Guo, S., Hu, X., Guo, S., Qiu, X., Qi, F.: Blockchain meets edge computing: a distributed and trusted authentication system. IEEE Trans. Ind. Inform. **16**(3), 1972–1983 (2020)
4. Neisse, R., Hernández-Ramos, J.L., Matheu, S.N., Baldini, G., Skarmeta, A.: Toward a blockchain-based platform to manage cybersecurity certification of IoT devices. In: 2019 IEEE Conference on Standards for Communications and Networking (CSCN), Granada, Spain, pp. 1–6 (2019)
5. Cui, Z., et al.: A hybrid blockchain-based identity authentication scheme for Multi-WSN. IEEE Trans. Serv. Comput. **13**(2), 241–251 (2020)
6. Thakker, J., Chang, I., Park, Y.: Secure data management in internet-of-things based on blockchain. In: 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, pp. 1–5 (2020)
7. Liu, M., Yu, F.R., Teng, Y., Leung, V.C.M., Song, M.: Computation offloading and content caching in wireless blockchain networks with mobile edge computing. IEEE Trans. Veh. Technol. **67**(11), 11008–11021 (2018)
8. Feng, J., Yu, F.R., Pei, Q., Chu, X., Du, J., Zhu, L.: Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach. IEEE Internet Things J. **7**(7), 6214–6228 (2020)
9. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)